Procedure: ***Designing the Software Architecture***	
Issue Date: January 9, 2001	**Procedure ID:** ***P-PE-020***
Supersedes: May 5, 2000	**Rev/Change 2.1**

1. **Purpose:** To design a Baseline Software Architecture.

2. **Applicability:** This procedure is applicable to all government and contractor personnel assigned to ATISD.

3. **Responsibility:** Software Manager

4. **Support:** Senior Systems Engineers

5. **Invoked By:**

Performing Baseline Requirements Analysis	P-PE-040

6. **Inputs:**

Agenda-	S-GP-131
SOW-	S-PM-010
Program Execution Plan-	S-PE-015
Requirements Specifications-	S-PE-187
Software Development Folder-	S-PE-275

7. **Outputs:**

Updated Requirements Traceability Matrix-	S-PE-190
Updated Software Development Folders-	S-PE-275
Updated Program Execution Plan-	S-PE-015
Software architecture design-	

8. **Procedures Invoked:**

Estimating Effort-	P-PE-175
Estimating Schedule-	P-PE-180

9. **External Procedures Referenced:** N/A

10. **Procedure Steps:**

a) The Software Manager initiates the software design effort. Schedule and conduct a design kick-off meeting with the Design Team to give an overview of the design phase. Review the schedule for completion of preliminary design and the responsibilities of each design team member. For preliminary design phase activities, discuss preliminary results obtained by the Design Team during the proposal and requirements analysis phases. Review the design methodology to be used.

b) The Design Team conducts an informal meeting to discuss and evaluate software and interface requirements, environment constraints, and design constraints. Determine which requirements can be met by both the operating system and the COTS/GOTS software products to be incorporated into the system. If an initial software architecture design exists, review it to determine any required changes or refinements. For projects involving enhancements or re-implementation of existing systems, analyze the existing software design.

c) The Design Team determines the architecture style(s) to be used in developing the architecture.

d) For requirements not fully allocated to the operating system, COTS, or GOTS products, the Design Team determines the top-level components or objects using the selected architecture style(s). For each software configuration item, illustrate the software architecture in a diagram that shows the organization and relationships between the top-level components/objects and any reused software or COTS products.

e) The Design Team performs functional or object decomposition. For 3GL non-object-oriented applications, divide the top level components into sub-components or units that can be represented by approximately 1,000 to 10,000 lines of code (LOC), i.e., data definition and executable lines of code (a smaller granularity can be used for smaller applications). For 3GL object-oriented applications, identify the services associated with each object. For 4GL applications, identify the screens and reports associated with each component. Update the software architecture diagram to show the decomposition.

f) The Design Team evaluates further options for reuse. Discuss, review, and finalize the software architecture as a team.

g) The Design Team records assumptions, rationale, and trade-offs in the applicable configuration item SDF. Place the software architecture diagrams in the applicable configuration item SDF(s). (This is the final step of this procedure for proposal-related software architecture designs.)

h) The Design Team reviews issues/factors that may affect the software design with the Project Manager and, when appropriate, with the client. Typical issues/factors to be considered include:

?? Intended type and scope of the software (e.g., mission critical, widely deployed system versus single site test bed support software)

?? Development requirements (e.g., cost, schedule, level of effort, support hardware/software, and training of client's staff)

?? Operational requirements (e.g., throughput, data rates, memory utilization, input/output channels, external data sources, operator-machine interface)

?? Impact on previously scheduled operational date for system

?? Impact of the current software development on existing system/software

?? Modify the design, if necessary to minimize risks and balance potentially conflicting issues/factors.

i) The Design Team updates the requirements database to show the requirement(s) that are either partially or fully implemented by each component. A component may fulfill more than one requirement. Alternatively, several components may be necessary to satisfy a requirement. However, every requirement must be traceable to at least one component and every component should partially or fully satisfy at least one requirement.

j) The Design Team updates the applicable component SDF to include component-specific design information, including a list of requirements implemented by the component. (If possible, generate this list from the requirements database.)

k) The Design Team submits the component design for peer review. (It is recommended that scheduled peer review(s) be conducted and that the review team include representatives from key functional areas on the project.) Resolve all comments. The Design Team updates estimates for effort (P-PE-175) and Schedule (P-PE-175) and place in the applicable SDF. Submit estimates to the Software Manager for incorporation into the project-level metrics database.

l) The Software Manager places the software architecture design under developmental configuration control.

m) For proposal-related software architecture, the Design Software Manager reaches a consensus with the supporting software engineers on software to be developed and the resulting decomposition.

11. Notes:

a) The initial software architecture defines a high-level software structure. This structure is decomposed to a level where major candidates for Commercial Off the Shelf (COTS) software products and reusable software can be identified and incorporated into the design. The software architecture design should include, for each software configuration item, a graphical representation of the design, a description of the top-level software components and the interfaces between the components, and the allocation of all software configuration item requirements to the components and the component interfaces. This design drives the initial software planning activities.

b) During the preliminary design phase, this procedure should be executed for each configuration item for which software is being developed.